CPSC 2221

Due Date: 19 Nov 2022, 11.59 PM

Marks – 50

Note for submission: Please submit a pdf containing the solution. No other type of submission is allowed. Paste your queries in text format with solution in legible screenshot must be pasted. Only one submission per student is allowed.

1. Answer the questions with respect to the table below.

users

id	name	age	gender	occ_id	city_id
1	John	25	M	1	3
2	Sara	20	F	3	4
3	Victor	31	M	2	5
4	Jane	27	F	1	3

occupation

id	name
1	Software Engineer
2	Accountant
3	Pharmacist
4	Library Assistant

City

id	name
1	Halifax
2	Calgary
3	Boston
4	New York
5	Toronto

(a) Create the three tables with their SQL queries and paste the code in text format below. Make sure while creating the tables: [5+2.5+2.5]

- Must have primary keys in all the tables.

- The attributes which can potentially be foreign keys must have NOT NULL Constraint.
- Gender attribute in users table should CHECK for ‘M’ or ‘F’.

Users Table Query :

```
CREATE TABLE Users (
        ID INT NOT NULL,
        Name Varchar(25),
        Age INT,
        Gender CHAR(1),
        Occ_ID INT NOT NULL,
        City_ID INT NOT NULL,
        CHECK (Gender = 'M' OR Gender = 'F'),
        PRIMARY KEY(ID),
        FOREIGN KEY(Occ_Id) REFERENCES Occupation(ID),
        FOREIGN KEY(City_ID) REFERENCES City(ID)
);
```

Occupation Table Query :

```
CREATE TABLE City (
        ID INT NOT NULL,
        Name VARCHAR(255),
        PRIMARY KEY(ID)
);
```

City Table Query :

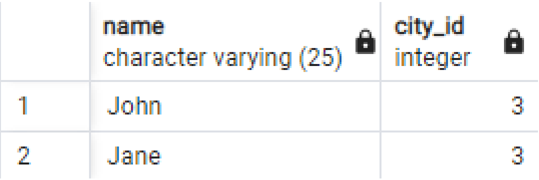
```
CREATE TABLE Occupation (
        ID INT NOT NULL,
        Name VARCHAR(255),
        PRIMARY KEY(ID)
);
```

(b) [10+5+5]

Write 2 different types of sql queries to find the users in city ‘Boston’. Write 2 types of queries, one using joins to find the answer and another using subqueries to find the same answer.

1) Join query
2)

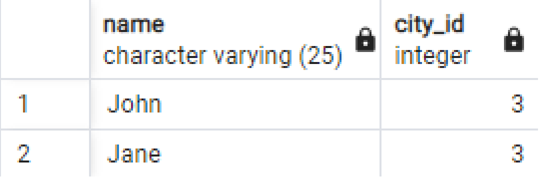
```
SELECT u.name, u.city_id
FROM Users u INNER JOIN City c
ON c.id = u.city_id
WHERE c.name = 'Boston'
```



	name character varying (25)	city_id integer
1	John	3
2	Jane	3

3) Subqueries

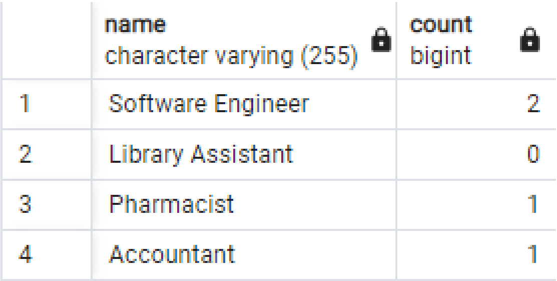
```
SELECT Name, City_ID
FROM Users
WHERE City_ID = (SELECT Id FROM City WHERE Name = 'Boston')
```



	name character varying (25)	city_id integer
1	John	3
2	Jane	3

Write sql query to find how many users are there per occupation.

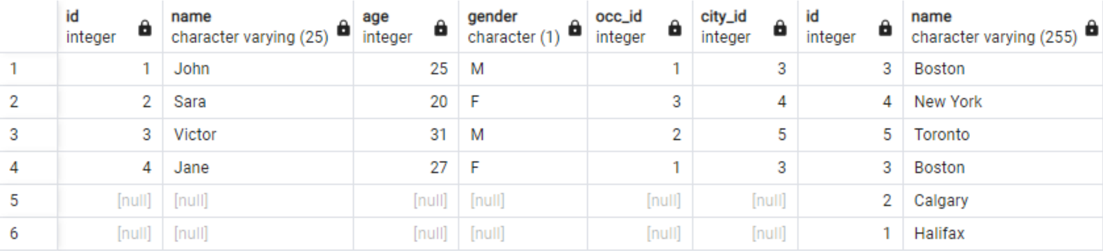
```
SELECT o.Name, COUNT(u.occ_id)
FROM occupation o LEFT JOIN Users u
ON o.id = u.occ_id
GROUP BY o.Name;
```



	name character varying (255)	count bigint
1	Software Engineer	2
2	Library Assistant	0
3	Pharmacist	1
4	Accountant	1

Perform full outer join between users and city.

```
SELECT *
FROM Users FULL OUTER JOIN City
ON Users.city_id = City.Id;
```

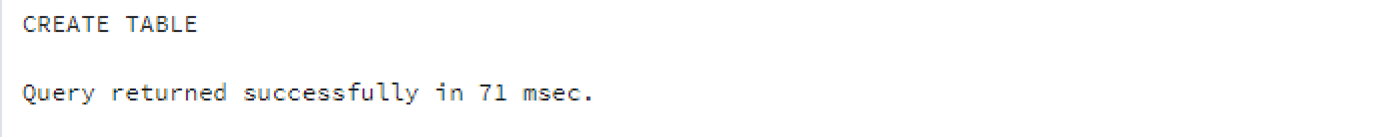


	id integer	name character varying (25)	age integer	gender character (1)	occ_id integer	city_id integer	id integer	name character varying (255)
1	1	John	25	M	1	3	3	Boston
2	2	Sara	20	F	3	4	4	New York
3	3	Victor	31	M	2	5	5	Toronto
4	4	Jane	27	F	1	3	3	Boston
5	[null]	[null]	[null]	[null]	[null]	[null]	2	Calgary
6	[null]	[null]	[null]	[null]	[null]	[null]	1	Halifax

(c) [2.5+2.5+5]

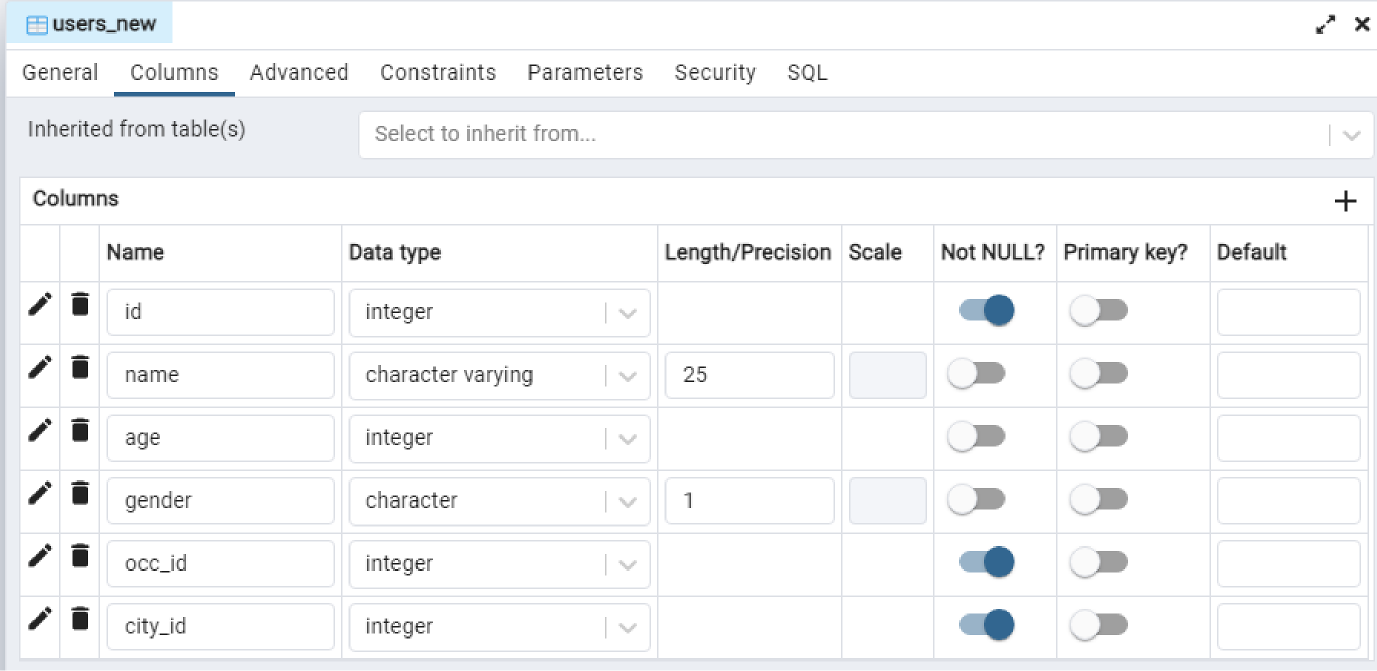
Write query to make a copy of ‘users’ table known as ‘users_new’ without Data.

```
CREATE TABLE users_new (LIKE users)
```



```
CREATE TABLE

Query returned successfully in 71 msec.
```



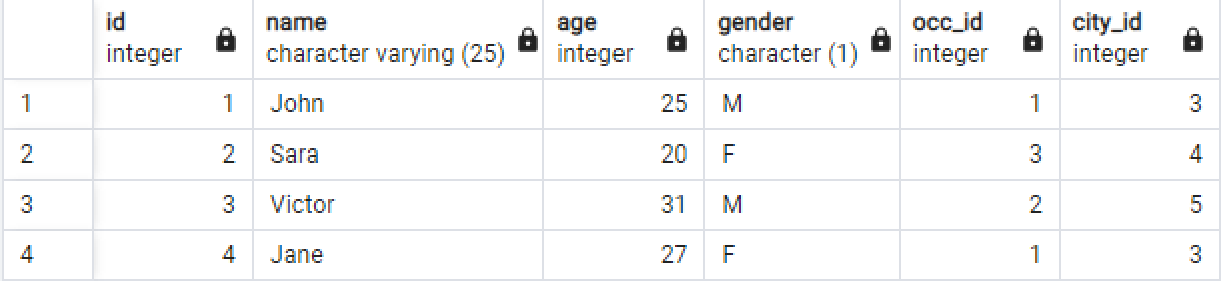
Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			●	○	
name	character varying	25		○	○	
age	integer			○	○	
gender	character	1		○	○	
occ_id	integer			●	○	
city_id	integer			●	○	

Write query to insert all columns of ‘users’ to the ‘users_new’.

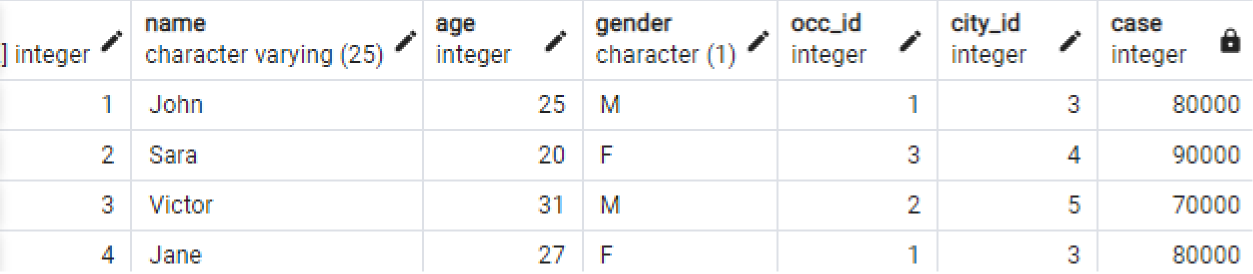
INSERT INTO users_new SELECT * FROM users;



	id integer	name character varying (25)	age integer	gender character (1)	occ_id integer	city_id integer
1	1	John	25	M	1	3
2	2	Sara	20	F	3	4
3	3	Victor	31	M	2	5
4	4	Jane	27	F	1	3

Write CASE query to add one more column with salary values to the ‘users’ table. Salary for Software engineer is 80,000, Accountant is 70,000 and Pharmacist is 90,000.

```
SELECT *,
    CASE
            WHEN occ_id = 1 THEN 80000
            WHEN occ_id = 2 THEN 70000
            WHEN occ_id = 3 THEN 90000
            ELSE 0
END
FROM users
```

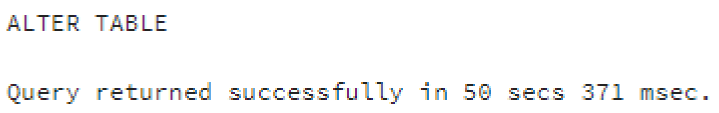


] integer	name character varying (25)	age integer	gender character (1)	occ_id integer	city_id integer	case integer
1	John	25	M	1	3	80000
2	Sara	20	F	3	4	90000
3	Victor	31	M	2	5	70000
4	Jane	27	F	1	3	80000

(d) [5+5]

Write query to add foreign keys constraints to ‘users’ table. Assuming you forgot to add it earlier

```
ALTER TABLE Users
ADD FOREIGN KEY (Occ_Id)
REFERENCES Occupation(id),
ADD FOREIGN KEY (City_Id)
REFERENCES City(id);
```

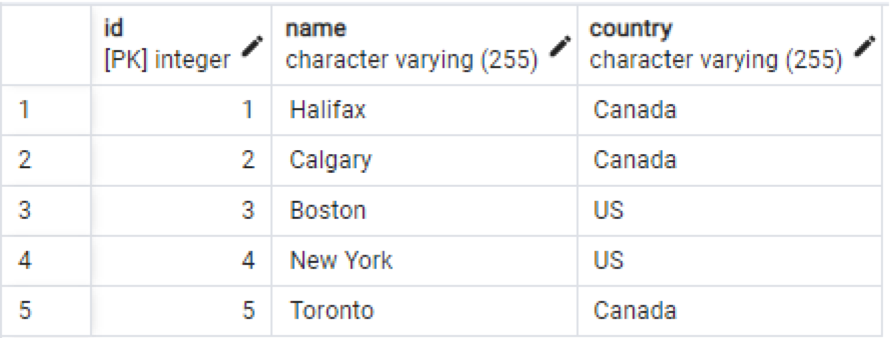


```
ALTER TABLE

Query returned successfully in 50 secs 371 msec.
```

Add country column to ‘city’ table. DEFAULT constraint must be used to add Canada as a default country for cities. [Use DEFAULT Constraint to default your country to Canada, that way you only have to write the countries for cities not in Canada, **Hint : remember ‘boston’ and ‘new York’ are cities in US, rest all are in Canada**, Use ALTER TABLE to add column and default constraint].

```
ALTER TABLE
City ADD Country VARCHAR(255)
DEFAULT 'Canada';
UPDATE City SET Country =
CASE WHEN Name = 'New York' THEN 'US'
WHEN Name = 'Boston' THEN 'US'
ELSE 'Canada' END;
```



	id [PK] integer	name character varying (255)	country character varying (255)
1	1	Halifax	Canada
2	2	Calgary	Canada
3	3	Boston	US
4	4	New York	US
5	5	Toronto	Canada